Terms used **instrument monitor memory access** Found **53,327** of **192,172**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 AccMon: Automatically Detecting Memory-Related Bugs via Program Counter-Based Invariants

Pin Zhou, Wei Liu, Long Fei, Shan Lu, Feng Qin, Yuanyuan Zhou, Samuel Midkiff, Josep Torrellas

December 2004 **Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture MICRO 37**

Publisher: IEEE Computer Society

Full text available: pdf(249.11 KB) Additional Information: full citation, abstract

This paper makes two contributions to architectural support for software debugging. First, it proposes a novel statistics-based, on-the-fly bug detectionmethod called PC-based invariant detection. The idea is based on the observation that, in most programs, a given memory location is typically accessed by only a few instructions. Therefore, by capturing the invariant of the set of PCs that normally access a given variable, we can detect accesses by outlier instructions, which are often caused by ...

2 Efficient and flexible architectural support for dynamic monitoring

Yuanyuan Zhou, Pin Zhou, Feng Qin, Wei Liu, Josep Torrellas

March 2005 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 2 Issue 1

Publisher: ACM Press

Full text available: pdf(524.21 KB) Additional Information: full citation, abstract, references, index terms

Recent impressive performance improvements in computer architecture have not led to significant gains in the case of debugging. Software debugging often relies on inserting run-time software checks. In many cases, however, it is hard to find the root cause of a bug. Moreover, program execution typically slows down significantly, often by 10--100 times.To address this problem, this paper introduces the *intelligent watcher (iWatcher)*, a novel architectural scheme to monitor dynamic executio ...

Keywords: Architectural support, dynamic monitoring, software debugging, thread-level speculation (TLS)

3 Cache Simulation Based on Runtime Instrumentation for OpenMP Applications

Jie Tao, Josef Weidendorfer

April 2004 **Proceedings of the 37th annual symposium on Simulation ANSS '04**

Publisher: IEEE Computer Society

Full text available: pdf(150.92 KB) Additional Information: full citation, abstract, index terms

To enable optimizations in memory access behavior ofhigh performance applications, cache monitoring is a crucialprocess. Simulation of cache hardware is needed in orderto allow research for non-existing cache architectures,and on the other hand, to get more insight into metrics notmeasured by hardware counters in existing processors.One focus of EP-Cache, a project investigating efficientprogramming on cache architectures, is on developingcache monitoring hardware to give precise informationabou ...

4 Compiler-directed run-time monitoring of program data access

Chen Ding, Yutao Zhong

June 2002 **ACM SIGPLAN Notices , Proceedings of the 2002 workshop on Memory system performance MSP '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available: pdf(1.40 MB) Additional Information: full citation, abstract, references, citings

Accurate run-time analysis has been expensive for complex programs, in part because most methods perform on all a data. Some applications require only partial reorganization. An example of this is off-loading infrequently used data from a mobile device. Complete monitoring is not necessary because not all accesses can reach the displaced data. To support partial monitoring, this paper presents a framework that includes a source-to-source C compiler and a run-time monitor. The compiler inserts ru ...

5 iWatcher: Efficient Architectural Support for Software Debugging

Pin Zhou, Feng Qin, Wei Liu, Yuanyuan Zhou, Josep Torrellas

March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual international symposium on Computer architecture ISCA '04**, Volume 32 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(314.11 KB) Additional Information: full citation, abstract, citings

Recent impressive performance improvements in computer architecturehave not led to significant gains in ease of debugging.Software debugging often relies on inserting run-time softwarechecks. In many cases, however, it is hard to find the root causeof a bug. Moreover, program execution typically slows down significantly,often by 10-100 times.To address this problem, this paper introduces the IntelligentWatcher (iWatcher), novel architectural support to monitor dynamicexecution with minimal overh ...

6 RaceTrack: efficient detection of data race conditions via adaptive tracking

Yuan Yu, Tom Rodeheffer, Wei Chen

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: pdf(321.34 KB) Additional Information: full citation, abstract, references, index terms

Bugs due to data races in multithreaded programs often exhibit non-deterministic symptoms and are notoriously difficult to find. This paper describes RaceTrack, a dynamic race detection tool that tracks the actions of a program and reports a warning whenever a suspicious pattern of activity has been observed. RaceTrack uses a novel hybrid detection algorithm and employs an adaptive approach that automatically directs more effort to areas that are more suspicious, thus providing more accurate war ...

Keywords: race detection, virtual machine instrumentation

7 Architectural support for performance tuning: a case study on the SPARCcenter 2000

A. Singhal, A. J. Goldberg

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94**, Volume 22 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

Latency hiding techniques such as multilevel cache hierarchies yield high performance when applications map well onto hierarchy implementations, but performance can suffer drastically when they do not. Identifying and reducing mismatches between an application and the memory hierarchy is difficult without insight into the actual behavior of the hardware implementation. We advocate the use of hardware event counters, as a cheap, effective and practical way to tune applications for a given hardwar ...

8 Identifying and Exploiting Spatial Regularity in Data Memory References

Tushar Mohan, Bronis R. de Supinski, Sally A. McKee, Frank Mueller, Andy Yoo, Martin Schulz

November 2003 **Proceedings of the 2003 ACM/IEEE conference on Supercomputing**

Publisher: IEEE Computer Society

Full text available: pdf(264.75 KB) Additional Information: full citation, abstract

The growing processor/memory performance gap causes the performance of many codes to be limited by memory accesses. If known to exist in an application, strided memory accesses forming streams can be targeted by optimizations such as prefetching, relocation, remapping, and vector loads. Undetected, they can be a significant source of memory stalls in loops. Existing stream-detection mechanisms either require special hardware, which may not gather statistics for subsequent analysis, or are limite ...

9 Track 4: reconfigurable computing (part 2): Owl: next generation system monitoring

Martin Schulz, Brian S. White, Sally A. McKee, Hsien-Hsin S. Lee, Jürgen Jeitner

May 2005 **Proceedings of the 2nd conference on Computing frontiers**

Publisher: ACM Press

Full text available: pdf(430.90 KB) Additional Information: full citation, abstract, references, index terms

As microarchitectural and system complexity grows, comprehending system behavior becomes increasingly difficult, and often requires obtaining and sifting through voluminous event traces or coordinating results from multiple, non-localized sources. Owl is a proposed framework that overcomes limitations faced by traditional performance counters and monitoring facilities in dealing with such complexity by pervasively deploying programmable monitoring elements throughout a system. The design exploit ...

Keywords: autonomous performance monitoring, performance analysis, reconfiguration

10 SMP system interconnect instrumentation for performance analysis

Lisa Noordergraaf, Robert Zak

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**

Publisher: IEEE Computer Society Press

Full text available: pdf(397.28 KB) Additional Information: full citation, abstract, references, citings, index terms

The system interconnect is often the performance bottleneck in SMP computers. Although modern SMPs include event counters on processors and interconnects, these provide limited information about the interaction of processors vying for shared resources. Additionally, transaction sources and addresses are not readily available, making analysis of access patterns and data locality difficult. Enhanced system interconnect instrumentation is required to extract this information.This paper describes in ...

11 Software visualization for specific domains: Interactive locality optimization on NUMA architectures

Tao Mu, Jie Tao, Martin Schulz, Sally A. McKee

June 2003 **Proceedings of the 2003 ACM symposium on Software visualization**

Publisher: ACM Press

Full text available: pdf(743.51 KB) Additional Information: full citation, abstract, references, index terms

Optimizing the performance of shared-memory NUMA programs remains something of a black art, requiring that application writers possess deep understanding of their programs' behaviors. This difficulty represents one of the remaining hindrances to the widespread adoption and deployment of these cost-efficient and scalable shared-memory NUMA architectures. To address this problem, we have developed a performance monitoring infrastructure and a corresponding set of tools to aid in visualizing and un ...

Keywords: NUMA Architectures, distributed systems, interactive locality optimizations, performance visualization

12 Nonintrusive precision instrumentation of microcontroller software

Ben L. Titzer, Jens Palsberg

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '05**, Volume 40 Issue 7

Publisher: ACM Press

Full text available: pdf(188.83 KB) Additional Information: full citation, abstract, references, citings, index terms

Debugging, testing, and profiling microcontroller programs are notoriously difficult. The lack of supporting software such as an operating system, a narrow interface to the hardware chip, and delicately timed sequences of code present significant challenges which can be exacerbated by the presence of additional debugging or profiling code. In this paper we present a solution to the precision instrumentation problem for microcontroller code that is based upon our open, flexible simulator framewor ...

Keywords: cycle-accurate, debugging, instruction-level simulation, instrumentation, monitoring, parallel simulation, profiling, sensor networks

13 Using Hardware Counters to Automatically Improve Memory Performance

Mustafa M. Tikir, Jeffrey K. Hollingsworth

November 2004 **Proceedings of the 2004 ACM/IEEE conference on Supercomputing**

Publisher: IEEE Computer Society

Full text available: pdf(152.84 KB) Additional Information: full citation, abstract

In this paper, we introduce a profile-driven online page migration scheme and investigate its impact on the performance of multithreaded applications. We use lightweight, inexpensive plug-in hardware counters to profile the memory access behavior of an application, and then migrate pages to memory local to the most frequently accessing processor. Using the Dyninst runtime instrumentation combined with hardware counters, we were able to add page migration capabilities to the system without having ...

14 SPiKE: engineering malware analysis tools using unobtrusive binary-instrumentation

Amit Vasudevan, Ramesh Yerraballi

January 2006 **Proceedings of the 29th Australasian Computer Science Conference - Volume 48 ACSC '06**

Publisher: Australian Computer Society, Inc.

Full text available: pdf(832.66 KB) Additional Information: full citation, abstract, references, index terms

Malware -- a generic term that encompasses viruses, trojans, spywares and other intrusive code -- is widespread today. Malware analysis is a multi-step process providing insight into malware structure and functionality, facilitating the development of an antidote. Behavior monitoring, an important step in the analysis process, is used to observe malware interaction with respect to the system and is achieved by employing dynamic coarse-grained binary-instrumentation on the target system. However, ...

Keywords: instrumentation, malware, security

15 Informing memory operations: memory performance feedback mechanisms and their applications

Mark Horowitz, Margaret Martonosi, Todd C. Mowry, Michael D. Smith
May 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 2
Publisher: ACM Press

Full text available: pdf(344.74 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Memory latency is an important bottleneck in system performance that cannot be adequately solved by hardware alone. Several promising software techniques have been shown to address this problem successfully in specific situations. However, the generality of these software approaches has been limited because current architecturtes do not provide a fine-grained, low-overhead mechanism for observing and reacting to memory behavior directly. To fill this need, this article proposes a new class ...

Keywords: cache miss notification, memory latency, processor architecture

16 Execution analysis of DSM applications: a distributed and scalable approach

Lionel Brunie, Laurent Lefèvre, Olivier Reymann
January 1996 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**
Publisher: ACM Press
Full text available: pdf(1.06 MB) Additional Information: full citation, references, citings, index terms

Keywords: distributed shared memory, monitoring, performance evaluation, program visualization

17 Cache performance analysis of traversals and random accesses

Richard E. Ladner, James D. Fix, Anthony LaMarca
January 1999 **Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms**
Publisher: Society for Industrial and Applied Mathematics
Full text available: pdf(1.07 MB) Additional Information: full citation, references, citings, index terms

18 A two-tier memory architecture for high-performance multiprocessor systems

T. M. Nguyen, V. P. Srini, A. M. Despain
June 1988 **Proceedings of the 2nd international conference on Supercomputing**
Publisher: ACM Press

Additional Information:

Full text available: pdf(1.38 MB) full citation, abstract, references, citings, index terms

Performance of high-speed multiprocessor systems is limited by the available bandwidth to memory and the need to synchronize write sharable data. This paper presents a new memory system that separates synchronization related data from others. The memory system has two tiers: synchronization memory and high bandwidth (HB) memory. The synchronization memory consists of snooping caches connected to a bus and is used to store synchronization variables such as locks and semaphores. The H ...

19 Shared-memory performance profiling

Zhichen Xu, James R. Larus, Barton P. Miller

June 1997 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '97**, Volume 32 Issue 7

Publisher: ACM Press

Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes a new approach to finding performance bottlenecks in shared-memory parallel programs and its embodiment in the Paradyn Parallel Performance Tools running with the Blizzard fine-grain distributed shared memory system. This approach exploits the underlying system's cache coherence protocol to detect data sharing patterns that indicate potential performance bottlenecks and presents performance measurements in a data-centric manner. As a demonstration, Parodyn helped us improve ...

20 Informing memory operations: providing memory performance feedback in modern processors

Mark Horowitz, Margaret Martonosi, Todd C. Mowry, Michael D. Smith

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.55 MB) Additional Information: full citation, abstract, references, citings, index terms

Memory latency is an important bottleneck in system performance that cannot be adequately solved by hardware alone. Several promising software techniques have been shown to address this problem successfully in specific situations. However, the generality of these software approaches has been limited because current architectures do not provide a fine-grained, low-overhead mechanism for observing and reacting to memory behavior directly. To fill this need, we propose a new class of memory operati ...

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	35	("5,325,430" "5,287,453" "5,175,856" "4,964,065" "5,388,268" "4,589,068" "5,237,691" "5,375,206" "5,647,056" "5,557,780" "5,675,798" "5,541,911" "5,632,022" "5,701,427" "5,602,729" "5,625,823" "5,636,376" "5,696,902").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 07:53
L2	4	l1 and allocated	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 07:54
L3	9	l1 and (allocated assigned)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:02
L4	8	l1 and (allocated assigned) and state	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 07:54
L5	24	monitor$3 with memory with allocation with state	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:17
L6	1876	memory adj2 access adj2 instruction$1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:17
L7	1924	memory near access near instruction$1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:18
L8	41	memory near monitor$3 near instruction$1	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:47

EAST Search History

L9	3	l7 and l8	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:18
L10	589	(717/127).CCLS.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:44
L11	386	717/130.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:48
L12	37	l10 and @ad < "19950531"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:48
L13	19	l11 and @ad < "19950531"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:48
L14	8	l12 and (memory near4 instruction$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:49
L15	9	l13 and (memory near4 instruction$1)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:50
L16	4	l14 and (allocated unallocated)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:51

EAST Search History

L17	6	l15 and (allocated unallocated)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/16 08:51